A Functional Optimization Based Approach for Continuous 3D Retargeted Touch of Arbitrary, Complex Boundaries in Haptic Virtual Reality

Yiwei Zhao
Stanford University
ywzhao@stanford.edu

Sean Follmer
Stanford University
sfollmer@stanford.edu

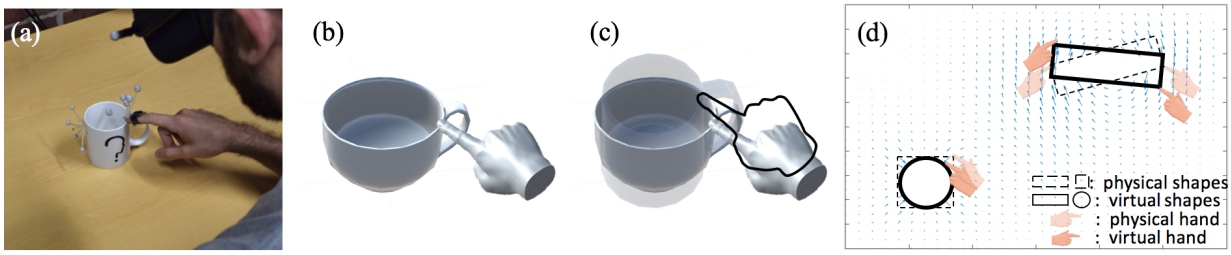

Figure 1. We present a new optimization based technique for 3D haptic retargeting of complex shapes. From left to right, a) A user interacts with a tracked physical prop (a coffee mug) while wearing an HMD, b) the virtual view of the user, showing a tea cup and retargeted hand position, c) the outline of the hand shows its position in the physical world, and the transparent coffee mug shows the shape and position of the real physical prop super imposed over the user's view, d) a 2D example of our method which shows how space is warped to retarget a square to a circle and move the position and orientation of another rectangle. The blue vectors represent the direction and magnitude of the spatial warping.

ABSTRACT

Passive or actuated physical props can provide haptic feedback, leading to a satisfying sense of presence and realism in virtual reality. However, the mismatch between the physical and virtual surfaces (boundaries) can diminish user experience. Haptic retargeting can overcome this limitation by utilizing visio-haptic effects. Previous investigations in haptic retargeting have focused on methods for point based position retargeting and techniques for remapping 2D shapes or simple 3D shape changes. Our approach extends haptic retargeting to complex, arbitrary shapes that provide a continuous mapping across all points on a boundary. This new approach also allows for multi-finger interaction. We describe a functional optimization to find the ideal spatial warping function with different goals: a maximum mapping smoothness, a minimum mismatch between the real and virtual world, or the combination of the two. We report on a preliminary user study of different optimization goals and elaborate potential applications through a set of demonstrations.

CHI 2018, April 21-26, 2018, Montreal, QC, Canada.

ACM ISBN 978-1-4503-5620-6/18/04...$15.00.
https://doi.org/10.1145/3173574.3174118

ACM Classification Keywords

H.5.1 Information Interfaces and Presentation: Multimedia Information Systems-Artificial, Augmented, and Virtual Realities;H.5.2User InterfacesHaptic I/O.

Author Keywords

Virtual Reality; Haptics; Haptic Illusion; Haptic Retargeting

INTRODUCTION

Recent advances in display technology and computational power have allowed virtual reality (VR) to once again seize attention both in academia and in industry. While today's VR displays have been developed to provide immersive visual experiences, further research is still required to support realistic haptic feedback for tangible interaction in virtual reality.

Many researchers have explored active kinesthetic and tactile haptic devices to render virtual forces and textures that a user perceives [5, 36]. However, these active techniques require complex hardware, and it is still challenging to render all of the different haptic sensations of an object, including shape, mass, and texture. Another approach is to use passive physical props to provide haptic feedback in VR. Passive props have physical properties that support many types of sensations, allowing users to directly touch them and leverage users' natural manipulation proficiency. Passive props, such as physical objects from a user's current physical environment, can be a useful haptic source for virtual objects [24, 7], but their shapes are fixed and cannot dynamically adapt to the content in virtual environment. Actuated physical props [18, 40, 45] can change their shape dynamically, but it is challenging and expensive to achieve high resolution displays. As a result, physical props

may not match well with the virtual object being manipulated, which can diminish user experience.

By utilizing visual dominance over the proprioceptive system, haptic retargeting can overcome this limitation by displacing user's hand to a virtual position. Several pseudo-haptics systems have been built to leverage previous investigations in haptic retargeting. Some of the research has focused on retargeting the position of objects for the purpose of reusing physical prop [7] or ergonomic interaction [39]. However, since only position is changed, the shape of the virtual object has to match that of the physical prop. Other researchers have investigated techniques for mapping between 2D cylinder-like shapes [9] or simple 3D space above a plane-like surface [28].

In this paper, we propose a new retargeting approach. Our method is fully 3D, which allows for arbitrary haptic interaction, where users can haptically explore the remapped object freely. Furthermore, our approach is capable of providing a continuous spatial warp to remap all points between two sets of complex boundaries (i.e. sets of shapes). The remapping of shapes is arbitrary (i.e. remapping of position, orientation, scale and shape), so we do not require the shape of the virtual and physical objects to match. Our approach also supports remapping multiple objects at the same time. Thirdly, our retargeting approach is based on warping the physical space point-wise, making it possible to support multi-finger interactions and even multi-user interactions. To achieve this, we use a functional optimization approach to find the ideal spatial warping function with the following objective: a maximum mapping smoothness, a minimum mismatch between the real and virtual world, or the hybrid of the two. This technique will also guarantee that the spatial warp mapping and the gradient of the mapping will be continuous, which leads to continuous speed and smooth motion through the space.

We present results from a preliminary user study of different optimization goals for different types of shapes. Our user study results confirm that our spatial warping techniques can help minimize user-perceived difference between the physical proxy and virtual shape. Our study finds that the objective functions for maximum smoothness warp and the hybrid model are better than the objective function for minimum mismatch warping for most cases in the re-targeting of small objects over small displacements.

In summary, our contributions are:

- The modeling and design of a 3D retargeting approach for complex continuous boundary (surface) remapping.
- The solving and analysis of this retargeting approach under different functional optimization goals.
- A preliminary user study evaluation comparing our method to a condition with no mismatch between the virtual and physical objects, which also shows the effect of different objective functions.

RELATED WORK

Haptics for VR Interaction

Several haptic systems have been developed to provide users with tangible interaction in virtual reality [23], which mostly rely on rendering kinesthetic forces to support manipulation. Many of them use either grounded [36] or hand mounted devices [2, 17]. Those active devices mainly aim to provide some specific type of haptic sensation such as single point kinesthesia [36], grasping [5, 17] or tactile feedback [49].

In contrast to active devices that create virtual haptic forces, passive haptic systems focus on using existing objects to provide all haptic sensations simultaneously. For some specific applications like scientific visualization [31] and medical data browsing [25], researchers have used 3D printed physical props with which users interact. Sheng et al. use a deformable physical prop for the creation and manipulation of conceptual 3D models [46]. Hettiarachchi et al. explore using physical objects from a user's current physical environment to provide the best-available props to map to specific virtual objects [24].

Encounter-type haptic devices achieve many of the similar goals as passive props in that a user does not need to wear or continuously hold the device, but rather comes into contact with it when interacting with a virtual object. In order to make the props or encounter type devices adapt their form to different virtual objects and render the shape dynamically, the concept of programmable matter has been proposed [48, 20]. Several actuated shape-changing interfaces have been created using different mechanisms, such as actuated curves [40], actuated pins arrays [18, 30], and constructive blocks [45, 44]. Researchers also developed robotic assembly systems to build low resolution haptic proxy objects for interactions [51].

However, for the existing physical proxy and encounter type haptic display methods, the shapes of passive props are either fixed, low in spatial resolution, or constrained in the shapes they can render. As a result, there could be a mismatch in shape between the real and the virtual objects, which may lead to poorer sense of presence [27] and decreased manipulation performance [32] during the interaction. Also, when the alignment is not close enough, a user's finger may float above or penetrate the virtual object, which is detrimental to the experience and sense of realism [13].

Visual Dominance in Haptic Perception

When there is a conflict between an observer's sense of vision and touch, vision is often strongly dominant, even without the observer's being aware of a conflict [19]. When viewing an object through a distorting lens, users preferred to think that the object was most similar to the distorted visual image, rather than the actual physical shape that they felt [43]. Based on the visual dominance on vestibular cues, researchers also scaled up or down the head rotations and translations, which has led to the creation of redirected walking technology [41].

When no tactile feedback is provided, the visual dominance effect is found to be very strong in haptic perception and propioception [15]. Even when the users are aware of the possibility of a mismatch, a 20 cm just noticeable difference (JND) between the real and virtual hand location has been reported. With normal and shear force feedback, researchers reported a 5.2 cm JND threshold for mismatch of fingertip position [35] and 3.2 cm for angular mismatch of finger flexion [37]. For mismatch of the direction of the force, Barbagli et al. [11]

reported the JND threshold to be 18.4° with haptics feedback alone, 25.6° for haptics plus congruent vision feedback, and 31.9° for haptics plus incongruent vision feedback.

Pseudo-Haptic Feedback

Researchers have leveraged the visual dominance over haptic perception to create a new technique to provide haptic feedback, called pseudo-haptic feedback. In this technique, the real haptic feedback provided is different from the simulated haptic sensation. This idea has also been used to modify the perception of shape, size, stiffness and texture [34]. Lecuyer et al. [33] showed that when a passive isometric input device is used together with visual feedback, it could provide the operator with a pseudo-stiffness of passive objects.

Among all the pseudo-haptic techniques, retargeting or redirecting, the process of warping the space and displacing the visual representation of the user's hand, are commonly used. Ban et al. used the pseudo-haptic effect to modify the identification of the shape of a curved surface; when the user touched it with an index finger, the virtual position of the hand was shifted. [8, 9, 10]. Kohli et al. explored methods that map many differently shaped virtual objects onto one physical object by warping virtual space [28], but this work mainly focused on mapping simple planes with different orientations or with small curvature changes. Their work also demonstrated that redirecting does not have negative effects on task performance for some tasks [29]. Azmandian et al. used haptic retargeting to repurpose a single physical prop in order to provide passive haptic sensation for multiple virtual objects [7]. In this approach, the virtual object has to match the physical object perfectly in shape and only the perceived location of the object is modified. Based on this, Cheng et al. augmented the existing Haptic Retargeting technique with an on-the-fly target remapping, which predicts users' intentions during interaction and consequently redirects their hand to a matching part of the passive proxy [16]. Murillo et al. used optimization based approaches to compute the best physical location to interact with each visual element, and then partitioned and distorted the space based on those mappings which allows for multi-object retargeting [39]. Using functional optimization techniques, our approach extends haptic retargeting to remap all points on a set of boundaries to another set of complex, arbitrary shapes, allowing for 3D, smooth hand displacement mapping.

Retargeting Methods

To warp the space and displace the visual representation of the user's hand, we need to find a map from the position of user's real hand/fingertip in real space to the position in virtual space. Different mapping methods have been used for different scenarios in the projects described above. In general, there are mainly four ways to generate the space warping map:

Linear Mapping

To retarget a point to a different position in a virtual environment, Azmandian [7] and Cheng [16] used a linear method to set the offset between the real and virtual hand. In this method, the offset position is gradually added to the position of the virtual hand as the user's hand starts moving from the initial position toward the physical object. In addition to world warping, they also utilized the method of body warping, which is used in redirected walking [41]. However for this method, only the perceived location of the object is modified and the shape of virtual object has to match that of the physical prop. Additionally, only one point can be retargeted at one time and the system needs to know [7] or predict [16] which point the user is going to touch.

Position Relation Based 2D Mapping

Ban et al. proposed a space warping procedure so that a simple static cylinder can be mapped to variety of shapes of rotationally symmetric bodies such as a cylinder and a barrel [8, 9, 10]. In their method, the distance between the fingertip and the boundary sides of the physical object are calculated, and based on this, the positional relation between the physical contour and fingertip is measured. Using the positional relation, the position of the virtual finger is determined. This is a 2D method where only the cross sections of the physical and virtual shapes are mapped. Additionally, users need to look down at the object and move their fingertip within the contour of the physical object. If the users look from different angle or move finger around the object, the method would fail.

Energy Optimization Mapping

Kohli et al. [28] used the Principal Warp method to find the spatial warp to map many differently shaped virtual objects onto one physical object. It guarantees that when a user's virtual finger touches the virtual boundary, their real finger also touches the physical boundary. The Principal Warp method is a widely used 2D method in image processing in which a 2D plane in 3D space is mapped to another 2D plane, with a set of controlling points and the minimum bending energy [14]. Kohli et al. [28] extended the Principal Warp method to a 3D version, but this method can only deal with simple plane cases with different orientations or with small curvature. Similarly in the graphics literature, researchers also use energy functions to find the best mapping function between two domains and best satisfy some user specified constraints. Ben-Chen et al. [12] used a functional optimization goal to find a "As-Rigid-As-Possible" harmonic deformation of the source region.

Mismatch and Ergonomic Optimization Mapping

Murillo et al. [39] also used the optimization method to find the spatial warp function. Their objective function is the ergonomic evaluation of the retargeted points as well as the mismatch between the real world and the warped world. Space is partitioned into several tetrahedrons and all their vertexes are retargeted based on the optimization goal. The displacement inside each tetrahedron is a combination of the displacement of its vertices. This method is fully 3D and can redirect multiple objects at the same time, effectively helping people reach the object out of range. But it can still only remap the position of the objects and cannot change their shapes.

To summarize, to our knowledge, our method is the first attempt to generate a continuous, fully 3D retargeting touch that can work for arbitrary and complex boundaries based on a functional optimization approach. Our method is also the first approach to optimize both mismatch and smoothness together.

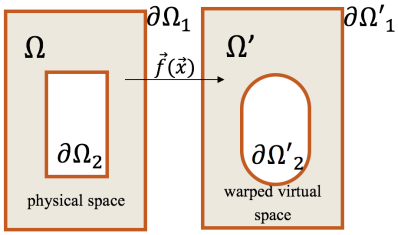

Figure 2. The mathematical description of the physical space, warped virtual space, boundaries and spatial warp function.

FUNCTIONAL OPTIMIZED 3D SPACE WARPPING

The goal of our functional optimized 3D retargeting method is to leverage the dominance of vision in haptic perception and find a way to warp the space surrounding the physical object. Using this spatial warp mapping, we can displace the user's hand so that the gap between the physical and virtual boundaries (surfaces) is eliminated. Then even if the passive and actuated physical props mismatch with the virtual object in shape, position, or boundary, they will still provide the illusion that the two boundaries match with each other.

To meet these goals, we designed our spatial warp approach to have the following properties:

- The mapping is fully 3D to support interaction in a Virtual Reality Environment. It also works for arbitrary pairs of complex surfaces in 3D spaces;
- The mapping is continuous everywhere in the space, so that the displaced virtual hand will not change its position suddenly when users move their hand;
- When users touches the boundary of the physical surface using their fingertip, their virtual fingertips are always mapped exactly to the boundary of the virtual surface so that the pseudo-haptic feedback will be provided by touching the boundary surface of the physical props;
- For our spatial warp map, the maximum displacement in the whole space is guaranteed to no more than that on the boundary. As long as the displacement of the retargeted object is controlled under the JND of mismatch, the displacement of virtual hand will always be less than the threshold.
- The optimal warp mapping is calculated through an optimization approach, with different objective functions: a maximum mapping smoothness, a minimum mismatch between the real and virtual world, or the combination.

Mathematical Description of the Spatial Warp Function

To define our spatial warp function, we firstly define the space surrounding the physical object to be the *physical space*, denoted as Ω, which is shown in Figure 2. Ω is a subset of the physical 3D world $\mathbb{R}^3$, namely $\Omega \subset \mathbb{R}^3$. The boundary of the physical space Ω is denoted as $\partial\Omega$, and there are two types of boundaries.

The first type of boundary is the outer boundary of the physical space, defined as $\partial\Omega_1$. No spatial warping will happen outside of this boundary, we only need to find the warp mapping inside this boundary. On this boundary, every point should be mapped to exactly the same position in virtual environment. This will guarantee that when users moves their hands/fingers across the outer boundary of the physical space, the virtual hands will not change position suddenly.

The second type of boundary is the surface which needs to be retargeted to a different shape, position, or orientation, defined as $\partial\Omega_2$. This boundary is the outline of the physical props and is the inner boundary of the physical space. We do not need to map the space inside the physical prop, since we assume the physical object is a rigid body, and the users' hands/fingers will never go into it.

The union of the two boundaries comprises the boundary of the physical space, namely $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$. Additionally, $\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$, which means the two types of boundaries will not intersect with each other.

Similarly to the space defined surrounding the physical object, we define the space surrounding virtual object to be the *warped virtual space*, denoted as Ω'. Ω'_1 is the outer boundary of the Ω'. Boundary Ω'_2 is the outline of the virtual shape, and the inner boundary of the Ω'.

In the next step, we define the *spatial warp* as a 3D mapping function $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x})] : \Omega \rightarrow \Omega'$, where f_i (i=1,2,3) is the scalar components of $\vec{f}$ in one of three axises of the space. The function $\vec{f}(\vec{x})$ is defined on the physical space and map any point in this space to the warped virtual world.

To guarantee the global continuity of the hand position mapping in the whole 3D space, we obtain one constraint of this function:

$$\begin{aligned} &\vec{f}(\vec{x}) = \vec{x} \text{ on } \partial\Omega_1 \\ &\partial\Omega_1 = \partial\Omega'_1 \end{aligned} \tag{1}$$

The goal of retargeting is to map the boundary of the physical prop to the boundary surface of the virtual object. Here we define our *retargeting goal* as $\vec{g}(\vec{x}) = [g_1(\vec{x}), g_2(\vec{x}), g_3(\vec{x})] : \partial\Omega_2 \rightarrow \partial\Omega'_2$, which is a 3D mapping from the surface of real object to that of virtual object. The spatial warp needs to meet the retargeting goal so that users' hands will be retargeted to the virtual surface when they are touching the surface of the physical props. According to this, we define the boundary condition as:

$$\vec{f}(\vec{x}) = \vec{g}(\vec{x}) \text{ on } \partial\Omega_2 \tag{2}$$

For many cases, we can obtain the retargeting goal easily by manually setting it. For example, if we want to map an object to different position, orientation, size or scale it in one dimension, we will automatically find the retargeting goal from the transformation matrix. For other cases, some other algorithm or techniques are needed to find it. We discussed this problem in more detail in the Retargeting Goal section.

Mathematical Description of the Optimization Goal

There are infinite number of spatial warps that can match the boundary condition, but we have to choose one 'best' function. Previous researchers in haptic retargeting have investigated several optimization goals to find a suitable function. Here we use two measurements to show how 'bad' a spatial warp is, which are set as the optimization goal to find the best spatial warp:

Firstly, like the Erg-O technique developed by Murillo [39], it is understood that the mismatch between the virtual world and the real world should be as small as possible. We define a functional $L_1^\alpha(\vec{f}(\vec{x}))$ to measure the mismatch induced by the spatial warp:

$$L_1^\alpha(\vec{f}(\vec{x})) = \int_\Omega \|\vec{f}(\vec{x}) - \vec{x}\|_2^\alpha d\vec{x} \tag{3}$$

Here $\|*\|_2$ is the l_2 norm of the vector and α is a integer that represents how much we care about the large mismatch. If α is large, it means we will penalize large error in position more.

Secondly, we don't want users to feel large changes or jumps in speed when they move their virtual hands. As in the bending energy based retargeting method [28], we hypothesize that the spatial warp would feel more natural if it is smooth. So we define a functional $L_2^\beta(\vec{f}(\vec{x}))$ to measure the unsmoothness of the displacement mapping, which is the mapping minus the hand position:

$$L_2^\beta(\vec{f}(\vec{x})) = \sum_i \int_\Omega \|\nabla\left(\vec{f}_i(\vec{x}) - x_i\right)\|_2^\beta d\vec{x}\,,\ i = 1,2,3 \tag{4}$$

Here β is a integer shows how much we care about the large speed change users feel during their hand movements. If β is large, it means we will penalize large change in velocity more.

So the total optimization goal can be defined as a linear combination of L_1^α and L_2^β:

$$L_{\alpha,\beta,\lambda}(\vec{f}(\vec{x})) = \lambda L_1^\alpha(\vec{f}(\vec{x})) + L_2^\beta(\vec{f}(\vec{x})) \tag{5}$$

Where λ is the parameter indicating the relative importance of the spatial-matching to the mapping-smoothness of the spatial warp. If λ equals 0, it means we only care about the smoothness for our optimization goal. On the contrary, when $\lambda \to +\infty$, it means we only care about the match between the physical space and virtual space for the optimization. We define λ to be the *relative optimization ratio*. Since the fundamental units of L_1^α and L_2^β are not the same, we can not add them up directly and need to divide all the physical quantities by the typical quantities to make them dimensionless. Here we assume all of the physical quantities have been nondimensionalized.

To summarize, in order to find the spatial warp, the optimization problem we need to solve is:

$$\begin{aligned} &\text{Minimize } L_{\alpha,\beta,\lambda}(\vec{f}(\vec{x})) \\ &\text{subject to: } \vec{f}(\vec{x}) = \vec{x} \text{ on } \partial\Omega_1 \\ &\qquad\qquad \vec{f}(\vec{x}) = \vec{g}(\vec{x}) \text{ on } \partial\Omega_2 \end{aligned} \tag{6}$$

Solving of the Optimization Problem

Since α and β are parameters we can adjust, we firstly discuss how we choose the proper α and β.

1. $\alpha = 1$, $\beta = 1$

With $\alpha = 1, \beta = 1$, it is shown that the solution for this problem is not always continuous and for some cases, is always zero inside the warped physical world and jumps to the retargeting goal on the boundary (see Appendix). This demonstrates that these are not good parameters for the measurement.

2. $\alpha = 2$, $\beta = 2$

Since $\int_\Omega \|\vec{f}(\vec{x}) - \vec{x}\|_2^2 d\vec{x} = \sum_i \int_\Omega (f_i(\vec{x}) - x_i)^2 d\vec{x}$, the optimization problem is equivalent to three separated optimization problems: For $i = 1,2,3$

$$\begin{aligned} &\text{Minimize } \lambda \int_\Omega (f_i(\vec{x}) - x_i)^2 d\vec{x} + \int_\Omega \left(\nabla(\vec{f}_i(\vec{x}) - x_i)\right)^2 d\vec{x} \\ &\text{subject to: } f_i(\vec{x}) = x_i \text{ on } \partial\Omega_1 \\ &\qquad\qquad f_i(\vec{x}) = g_i(\vec{x}) \text{ on } \partial\Omega_2 \end{aligned} \tag{7}$$

We apply the Calculus of Variations method to these functional optimization problems to transform them into three Partial Differential Equations (PDEs) [47]. The optimization problem above is equivalent to the following PDEs for $i = 1,2,3$

$$\begin{aligned} \Delta f_i(\vec{x}) &= \lambda(f_i(\vec{x}) - x_i) \text{ in } \Omega \\ \text{subject to: } f_i(\vec{x}) &= \vec{x} \text{ on } \partial\Omega_1 \\ f_i(\vec{x}) &= g_i(\vec{x}) \text{ on } \partial\Omega_2 \end{aligned} \tag{8}$$

Since each of the PDEs is a linear, second order PDE with the the Dirichlet boundary condition, the optimization problem can be solved in several ways using numerical methods [38]. Choosing a higher α and β will achieve a similar effect (avoiding the large mismatch but tolerating the small), however will become very complex to solve. So the following discussion we will keep using the setting of $\alpha = 2$, $\beta = 2$.

Result Discussion

From the properties of the linear second order PDE [47], we can obtain the mathematical properties of our spatial warp solution:

- On the surface of the physical space, the boundary condition is satisfied so retargeting goal can be exactly reached.
- The spatial warp solution is continuous everywhere in the physical space. Also the constrained design of the outer boundary ensures the hand position mapping to be continuous globally.
- Using Hopf Maximum Principle [26], we proved that the max mismatch in physical space will not be greater that on the boundary. This means as long as the maximum mismatch of the boundary of the retargeting goal is controlled to under the JND, the max mismatch of our spatial warp will also be controlled under that threshold.

1-D theoretical solution

We can easily solve the problem in the 1D case. Consider the domain $\Omega = [0,1]$ and boundary condition $f(0) = 0$ and $f(1) = k$, we find the solution to be:

$$f(x) = \frac{k-1}{e^{\sqrt{\lambda}} - e^{-\sqrt{\lambda}}}(e^{\sqrt{\lambda}x} - e^{-\sqrt{\lambda}x}) + x \tag{9}$$

when $\lambda \to 0$, the $f(x)$ becomes $f(x) = kx$, which is just the linear method.

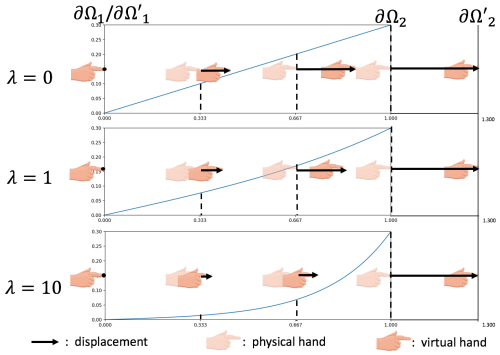

Figure 3. 1D results of our method for retargeting to a different line length ($\partial\Omega_2$ to $\partial\Omega'_2$). When $\lambda = 0$, the mapping is linear and the offset between the physical hand and the virtual hand increases linearly with displacement. As λ increases the mapping becomes more exponential.

We plot our 1D results in Figure 3, with different relative optimization ratios λ to see its effect. When λ is zero, we only care about the smoothness of our spatial warp, and the result is a affine function; the change of our mapping (gradient) is uniformly distributed in the physical space. If a users move their hands at a constant speed, their hands will also move uniformly, meaning the mapping is pretty smooth.

Then we change the λ to be 1, meaning we care equally about the smoothness and mismatching. The results shows when the hand is far away from the retargeted object, the mismatch is small. Then the mapping increases exponentially to the virtual shape to meet the boundary condition. If we increase the λ further, we can see the jump happens more suddenly.

2D numerical solution

2D Helmholtz equations do not have theoretical solutions for arbitrary boundaries, but we can solve them quickly numerically. Here, we used the MATLAB PDE solver to get the numerical solution [3] which takes 2.334 secs on a 2015 Macbook Pro. The mesh contains 1975 points and solution is obtained through the finite element approach.

Figure 4 is a solution example where we map the sphere on the left side to be a square. We also map the orientation of the right rectangle. This result is obtained using $\lambda = 0$. Figure 4(a) shows the length of gradient of out method, i.e. speed change of the hand displacement. We can see that the large speed changes mainly happen surrounding the boundary of the remapped shapes, especially at the corners. But in most part of the physical space, the mapping is smooth. Figure 4(b,c) show the displacement of hand in x and y axis. It can be shown that the maximum mismatch only happens at the boundary, as we proved mathematically.

This 2D case shows that our method can smoothly warp the physical space for different retargeting goals at the same time. It also shows our method supports multi-object retargeting with different types of retargeting goals (position, orientation and shape).

3-D numerical solution

We can also solve the 3D Helmholtz equation numerically to obtain the 3D spatial warp. Here we used COMSOL PDE

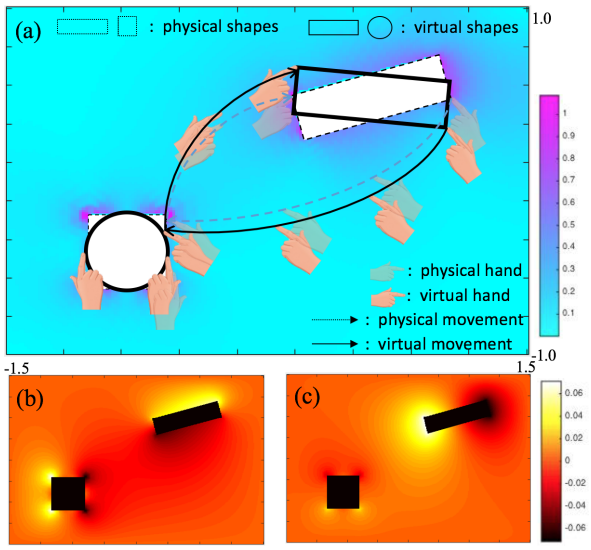

Figure 4. 2D results of remapping a square to a circle and changing the orientation of a rectangle. (a) Euclidean norm of the gradient. (b) x displacement. (c) y displacement.

solver [1]. In Figure 5, we show three types of retargeting goals for the 3D shape: 1) we stretched a sphere along one axis and scaled it up in this dimension by 1.5 times in Figure 5(a), 2) we change the size of a cube to be 1/4 of its original volume in Figure 5(b), 3) we rounded the corner of a square from a fillet radius of 10mm to 30mm in Figure 5(c). We used $\lambda = 0$ for the results in Figure 5. From the result, we can see that:

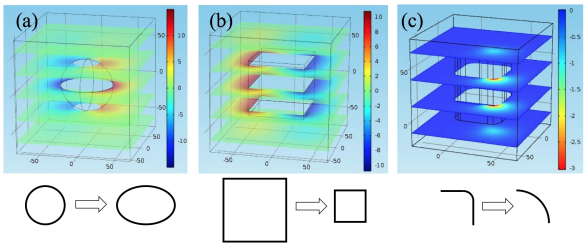

Figure 5. 3D results for spatial warping of different remapped objects. Figures show 2D cross sections of the x displacement. (a) A sphere scaled along one axis by 1.5x; (b) A cube mapped to be 1/4 of its original size; (c) We rounded the conner of a square from 10mm to 30mm.

- As we proved mathematically before, the maximum of mismatch will only happen at the boundary.
- For the stretch and size cases, the displacement distributed uniformly surround the boundary, but for the corner case, the displacement happens mainly surround the corner.

DEFINING RETARGETING GOALS

In the section above, we assume the retargeting goal, namely the boundary condition of our optimization problem, is already known. In many cases, we can define the retargeting goal easily by directly setting it. For example, if we want to map an object to a different position, orientation, size, or scale it in one dimension, we will automatically know where the point on the boundary should be mapped to through a transformation matrix.

For the cases of mapping two shapes, there is no perfect way to define the retargeting goal given two arbitrary shapes and the method we need may vary in different situations. In this section, we discuss how we can represent and solve the retargeting goal for shape changing cases. Here we assume that the two shapes are in the same position and orientation.

Shape Change as Texture Displacement Mapping

Here we introduce the *Shape Changing Texture Map*, which is a representation of the retargeting goal for shape changing cases. Since the retargeting goal is a 3D function defined on the surface of the physical object, we can show the displacement map $(\vec{g}(\vec{x}) - \vec{x})$ as a texture of the physical surface. In Figure 6, we remap a cube to be a sphere and the color's RGB channels represent the displacement values along the x, y, and z axes, respectively. This method provides a simple way to represent the retargeting goal using displacement mapping techniques often used in Computer Graphics.

An ideal mapping should map every point on the physical prop to a unique point on the virtual object. Also, since we do not want users to feel any jump when they move their fingers along the surface, the function and its inverse should be smooth.

Figure 6. An example of a Shape Changing Texture Map, where the surface of a cube is mapped to the surface of sphere. Color represents the displacement values.

Methods to find the Shape Changing Texture Map

For surfaces which can be described by a 2D function, we can project the surface to onto a plane and bi-map two points if they share the same projected position, which is shown in Figure 7(a). This method can be extended to the condition in which a boundary is composed of several surfaces like this and map each of those separately. For the boundary of convex object in Figure 7(b), we can use the spherical coordinate system to describe the surface and bi-map two points if they have the same azimuthal angle and polar angle.

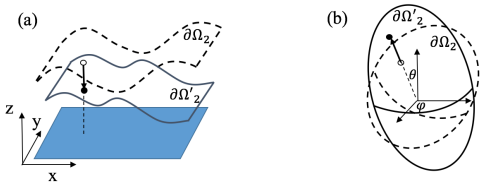

Figure 7. Finding the Shape Changing Texture Map using either a plane projection in (a) or a spherical coordinate system (b).

For arbitrary complex shapes, if we want to find a smooth invertible one-to-one function with a smooth inverse as well, we can apply a technique called Area Preserving Mapping [21]. This method can generate a diffeomorphism between two arbitrary surfaces as long as their topologies are the same, with an optimal total transportation. Also under this map, area will be preserved; as long as two pieces of surface have the same area before mapping, the areas will still be equal after mapping. In Figure 8, we show how we map a bunny to its low resolution prop using the Area Preserving Mapping. Though this method can work for arbitrary shapes, its limitation is that it ignores all the local curvature information; a corner may be mapped to be flat, which can be a detriment to the user's haptic perception. In summary, finding a general method for defining mapping goals for arbitrary shapes which also match the local curvature remains a difficult challenge that we will discuss in the Future Work section.

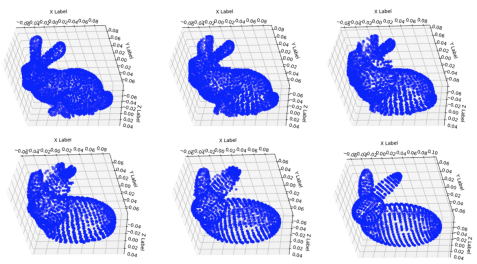

Figure 8. Finding the Shape Changing Texture Map using an Area Preserving Map: This example shows how we gradually map a bunny to its more simple physical proxy.

IMPLEMENTATION

In order to apply these methods, the following procedures are performed to generate a spatial warping retargeting:

Firstly, based on the application scenario, we defined which boundaries need to be remapped and which is the the outer boundary of the warped physical space. Then the 3D model of the physical object together with the outer physical boundary is built and imported to a PDE solver. For this research, we used COMSOL [1].

Secondly, we set the boundary condition for each of the virtual boundaries, i.e. the retargeting goal. If the boundary we want to remap is to a different position, orientation, size, or scale in one dimension, we can directly input the boundary condition. If the boundary we want to remap is to a different complex shape, we use the methods we mentioned in Shape Changing Texture Section to calculate the displacement on the boundary. Then we solve the PDE and output the solution as a 3D look up table from position to displacement, at a chosen arbitrary resolution.

Finally, the virtual environment and virtual object are programmed using the Unity game engine. The solution of PDE (look up table) is also imported to Unity. A OptiTrack [4] system is used to track the position of the hand and the physical prop. For each frame, we obtain the relative position of the hand in the warped physical space, look up the table to find the corresponding displacement, then place the virtual hand. The Oculus Rift was used as the Head-Mounted Display (HMD).

USER STUDY

We designed a user study to investigate if our method can provide users with a satisfying sense of realism by using physical props, even though their boundaries differ from the virtual object. Users were asked to touch and feel a physical object

using a single finger while wearing a HMD and viewing a virtual hand and a virtual object in VR. After this interaction, we asked three questions to evaluate the user's feelings about realism. To compare the effectiveness of relative optimization ratios, we also used different ratios to investigate which optimization goal most effectively creates a haptic illusion.

Participants

The experiment was conducted with 10 participants (2 female and 8 male, between the ages of 21 and 27). The experiment took approximately 65 minutes per participant, and each participant was tested individually.

Selecting Shapes and Parameters

We explored three categories of shape change: *stretch*, *size* and *corner*, shown in Figure 9(a). Stretch refers to changing the scale of an object along one axis. Size refers to changing the scale of an object along all axes with same ratio. Corner refers to changing the fillet radius of a edge of an object. For each category, we select three different versions of a shape with different amounts of stretch, size or radius respectively. For the stretch category, the shapes are ovals with a diameter of $70mm$ in two axes and $7cm$, $8.5cm$ and $10cm$ respectively in the third axis. For the size category, the shapes are cubes with volume of $125cm^3$, $250cm^3$ and $500cm^3$. For the corner category, the shapes are cuboids with one edge filleted. The fillet radius are $1cm$, $2cm$ and $3cm$, respectively. The magnitudes of the shape changes were chosen to be under the JND for fingertip mismatch with force feedback [35].

Three different relative optimization ratios λ were used, which were set to be 0 (λ_0), 1 (λ_1) and 10 (λ_{10}). When $\lambda = 0$, it means we only care about the smoothness of our mapping. When $\lambda = 1$ it means we care about the smoothness and mismatch equally for our mapping. λ can not be set to infinity since we need to guarantee continuity of the mapping, so we chose the third value ($\lambda = 10$) which we empirically determined to be large enough for the resolution of the PDE solver we used (COMSOL).

For each of the trials, a virtual object was rendered in the virtual environment and a physical prop was provided for haptic feedback. The shape of the virtual object and physical prop were selected from the same category such that there were 9 total physical-virtual shape pairs for each category. No mapping was added between identical physical and virtual shapes. Otherwise, we added our spatial warp to the hand, using one of the three λs in each trail. Therefore there are a total of 3 categories $\times$ (A_3^2 different-shape pairs $\times$ $3\lambda + A_3^1$ same-shape pairs) = 63 trials for each participant. In total, we conducted 594 trials.

Environment Implementation

The Oculus Rift was used as the Head-Mounted Display (HMD). A platform with retro-reflective markers was built to track the position of the physical props. The top plate of the platform shown in Figure 9(c) was removable so we could switch the physical prop easily. The relative displacement and rotation between platform and object were known from calibration. Shown in Figure 9(b), retro-reflective markers were also attached to the user's fingertip to get its 3D position and orientation using OptiTrack IR tracking system [4]. This 3D position and orientation was mapped to the index fingertip of a rigid 3D hand model in the virtual environment. The virtual environment was programmed in Unity.

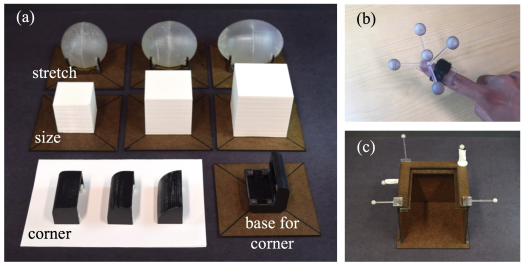

Figure 9. Study setup: a) the three different physical prop conditions and the three different versions of each, for corner cases, a switchable base is designed to change the edge fast, b) a user's finger with a retro-reflective marker for finger tracking, c) the platform to hold and track the different physical props.

Procedure

At the beginning of the study, participants were told about the tasks and questionnaires they need to answer. All users were informed that their hand position, hand movement and objects' shapes might be different from what they see in VR. The physical objects used in the study were not shown to the users before or during the task. During each trial, the participant was presented with one of the physical-virtual object pairs. The participant was then asked to move her/his finger to touch the object's surface. To include more complex interaction over a larger area, the participant was required to tap the object at least 3 times for each trial. Beyond this, there were no constraints on the haptic exploration by participants. After 15 seconds, the participant was asked to answer three questions:

1. How well did your virtual finger match the position of your real finger?
2. How well did your virtual finger match the movement of your real finger?
3. How well did the virtual shape match the physical shape?

The participant then responded with a score from 1 to 5 for each of the questions; 5 signified matching perfectly while 1 signified they did not match at all.

We hypothesized that:

- Since the spatial warp guarantees that the maximum mismatch is less than the mismatch on the boundaries, which were chosen to be under the JND, participants will not notice the positional displacement of their virtual finger (H1).
- Since the spatial warp guarantees continuity, therefore no sudden jumps in movement will occur, we hypothesize participants will not notice any movement mismatch (H2).
- The spatial warp will cause participants to not notice the shape discrepancy between the virtual and the physical object (H3).
- The Changing of λ will affect participants' perception about the mismatches (H4).

Results

We use the post-hoc analysis to filter out the outliers beyond the mean$\pm$2 standard deviations. The statical results of users' perception about virtual hand position, movement and virtual shape under different situations are shown in Figure 10. We use Wilcoxon signed-rank test [50] to analyze the difference of rating using different λ. The t-test is used for the comparation between the cases of same shapes cases and different shapes cases. All the significant differences are labeled with p-values.

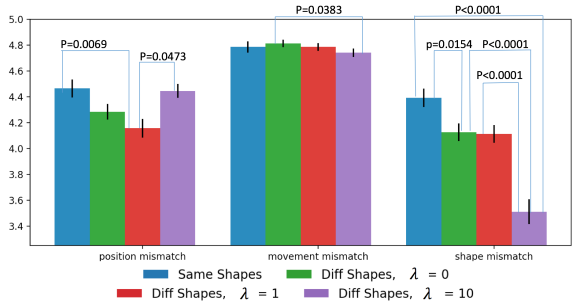

Figure 10. Results from participant responses about mismatch. Four conditions are compared here: Same physical and virtual shapes, different physical and virtual shapes with $\lambda = 0, 1, 10$. All the significant results are marked with a p value. Error bars represent standard error.

Position Mismatch Perception

The results shown in Figure 10 revealed no significant difference between the same shape cases and different shape cases with $\lambda = 0$ spatial warping (p>.05). However, when $\lambda = 1$ was used, users' perceived mismatch between virtual and real finger position increased (p=.0069). When λ increases further and more weight is given to the mismatch term, the rating for finger position rises again: $\lambda = 10$ mapping was better than $\lambda = 1$ (p=.0473) for perceived finger mismatch.

In Figure 11, we also show that under the $\lambda = 0$ setting, for different categories of retargeting, users have a similar rating between same shape cases (where the virtual object matches the physical prop) and different shape cases (where the virtual object does not match the physical prop).

Movement Mismatch Perception

Similar high rating was reported for movement mismatch perception. Figure 11 also shows that for different categories of retargeting, users all perceived that the virtual movement was identical to the actual finger's movement.

Shape Mismatch Perception

For the perception of shape mismatch, results show even with the retargeting method of $\lambda = 0$, when the virtual shape differs from the physical shape, users rated the virtual to physical shape matching significantly lower (p=.0154). But overall, the rating was not too low, with an average 4.13. No significant difference is reported between $\lambda = 0$ and $\lambda = 1$, but $\lambda = 10$ method is significantly worse than the others (p<.0001).

For different categories, it is shown that our method worked well with stretch and size, but was relatively ineffective for the corner cases. Participants perceived the large change of surface curvature and reported a low rating, *"I can feel by my finger that there was an edge, even though I couldn't see them in the VR."*(P2)

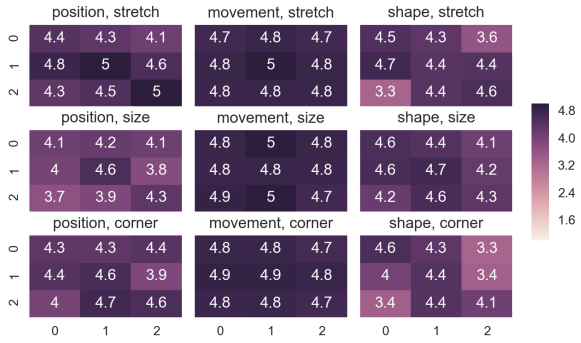

Figure 11. The average ratings for position, movement and shape perception matching, with different shape categories (from top to bottom rows: stretch, size, corner radius). The cases on the diagonal of each matrix show the results for same physical and virtual shapes while the others show the results with $\lambda = 0$ spatial warping when physical and virtual shapes are different.

Discussion

Overall, results show that by making the spatial warp mapping continuous and controlling the maximum position difference, the user reported scores for hand's position and movement are generally high. In terms of shape perception, our method also worked well for the cube and sphere cases, resulting in high reported scores. The corner condition has a rounded corner with radius range between 1 to 3 cm, so there was no sharp edge. Though, as shown in Figure 11, when there is a large mismatch of this corner radius, the average perceived shape match is lowest. This likely means that when high frequency shape information does not match between the prop and the virtual object, there will be a conflict in the user's perception.

The change of λ also affected users' feeling about the virtual and physical mismatch. No significant difference was reported between the $\lambda = 0$ and $\lambda = 1$ methods. But when λ was increased dramatically, meaning we care more about minimizing the total displacement of the virtual hand, users felt their virtual hand position better match their own. However when λ is too big, the perception rating for shape-matching got worse. Therefore, ideally we recommend using a λ between 0 and 1. Finally, the change of λ did not significantly affect the perception of movement mismatch. We think this may be because in this study, sizes of warped physical space were small (less than 25cm) and no large movement was seen in the space. The effect of λ on movement perception in large scale retargeting remains an interesting topic that is discussed in the Future Work section.

LIMITATIONS AND FUTURE WORK

For our spatial warping method, the retargeting goal (i.e. the boundary conditions) is an input. In this paper we provide several methods to find the diffeomorphism between two arbitrary shapes, in cases where we cannot obtain it directly or easily. But for most of those methods [21, 22], the local curvature information is lost, which has been shown to be important for user's perception of shape [11, 42]. A general

strategy to find the retargeting goal between two shapes, with the maximum surface curvature match, would be an important topic to investigate in future work.

In our user study we chose some specific objects and shape transformations (stretch, scale, corner radius) to investigate the effectiveness of our method. However, even though we can control the maximum displacement mismatch under the JND threshold and remap arbitrary shapes, we can not always provide all the curvature cues using physical props. Future work should investigate the JND threshold of the shape change and local surface curvature change with spatial warping, under different situations such as different haptic exploration methods and different interaction times.

The results from our user study did not reveal a significant effect of λ on movement mismatch perception. We believe this is because the scale of movement in our study was relatively small. Exploration of the effect of λ on large scale movement is also an interesting area of future work.

Besides the limitations described above, we also find the following topics interesting to explore in future work:

Combination with Encounter-type Haptics Systems In this paper, we assume the shape/boundary of the physical props is fixed. For some other encounter-type haptics systems, like 2.5D shape displays [18], robotically assembled displays [51] and constructive blocks system [45], the surface of the physical prop is actuated. Other systems for encounter-type haptics can change the physical position of a prop using robotics. It would be interesting to combined the spatial warping method with the design and control of shape-changing interfaces, to make them more expressive and low cost. In addition, an optimization based approach to space warping with actuated props which takes into account the time and energy needed to physically move an object or physically change shape as another objective function could be explored.

Multi-finger Retargeting Beyond mapping only a single fingertip, we could extend this to support whole-hand or multi-finger retargeting. A naive approach would be to map every point of the hand based on the spatial warp mapping we obtained. However, since our spatial warp is not a rigid transformation, this approach will cause the hand distortion and potentially break the illusion. An alternative approach is to retarget the hand pose: we can model our hand as a linked skeleton with five end-effectors, i.e. fingertips. Since we assume the interaction mainly happens on the fingertips, we could only displace the position of fingertips. Then in order to get the overall hand pose inverse kinematics algorithm such as FABRIK [6] could be used to calculate the angles of joints, which would be used to render the virtual hand. Additional constraints including max finger joint rotation mismatch and positional mismatch, could be included to improve this technique, making the overall optimization problem an interesting future work topic.

Physical Space Overlapping When the Physical Spaces of two objects overlap with each other, the spatial warp function will not be unique, so we need a way to combine two warps together. One naive method would be combine the two boundaries of two objects as the new boundary condition and solve the new optimization problem and obtain the new spatial warp which can satisfy two objects' retargeting goals. However, this method may be computationally-expensive; whenever the related position of two objects changes, we need to recompute the whole PDE. A fast algorithm to combine two spatial warps that also satisfies the overall boundary condition and maximizes the smoothness is a possible interesting future topic.

Effect of Physical Space Size We haven't yet explored the effect of the size of the exterior boundary volume; currently in the paper, the volume is heuristically set to be 10 times of the bounding box of the interior object. But empirically when volume is large, the mapping becomes smoother. However, larger volumes require more time to compute and more disk drive space to store, so there are some trade-offs. A more rigorous study on the effect of exterior bondary volume size on user's perception and how to properly set the bondary volume size remains a topic to explore.

Minimizing Warp Close to the User's Body When the hand is close to the user's body, proprioception may reveal the existence of large warp and diminish the sense of realism. However when the object surfaces are away from the body, user is much less sensitive to the existence of the warp. An interesting future work would be combining this effect with the optimization term. For example we can add spatial weight to the integration of the loss and lower the requirement of L_1 for the space which is not close to the user. This strategy may have effect on both the perception and computional complexity.

CONCLUSION

In this paper we introduced a new technique for continuous haptic retargeting of complex arbitrary 3D shapes based on a functional optimization. This optimization can have three goals, maximizing the smoothness (gradient of the mapping), minimizing position mismatch, or the combination of the two. We also described techniques for defining the boundary conditions for this optimization. From our preliminary user study we have shown that this technique can be useful when the mismatch between the two boundary conditions (surfaces) are under the Just Noticeable Threshold. We believe these techniques can be readily applied to haptic retargeting of physical props in virtual reality, and we hope that this will enable further study and development of haptic retargeting techniques.

APPENDIX A: DISCONTINUITY FOR $\alpha = 1, \beta = 1$

We show the discontinuity by using a 1D example: For retargeting 1-d problem, let $f(x) = g(x) - x$. Since $f'(x) > 0$, we get $L_2^1 = \int_0^1 \|f'(x)\| = \int_0^1 f'(x) = f(1) - f(0)$, which is a constant. However, since the function $x + (k-1) \times \delta(1)$ has the minimum loss for L_1^1, we get this discontinuous function to be the optimized solution.

ACKNOWLEDGMENTS

The authors would like to thank Mathieu Le Goc, Alexa Siu, Eric J. Gonzalez, Ye Wang, Chi Zhang and Ziyue Wang for their kind helps about the paper.

REFERENCES

1. 2017. COMSOL PDE Solver. https://www.comsol.com/. (2017).
2. 2017. Cybergrasp, cyberglove systems inc. http://www.cyberglovesystems.com/cybergrasp/. (2017).
3. 2017. Matlab PDE Solver. http://www.mathworks.com/help/matlab/math/partial-differential-equations.html. (2017).
4. 2017. OptiTrack tracking system. http://optitrack.com/. (2017).
5. Manuel Aiple and André Schiele. 2013. Pushing the limits of the CyberGrasp for haptic rendering. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 3541–3546.
6. Andreas Aristidou and Joan Lasenby. 2011. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models* 73, 5 (2011), 243–260.
7. Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. 2016. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1968–1979.
8. Yuki Ban, Takashi Kajinami, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2012a. Modifying an identified angle of edged shapes using pseudo-haptic effects. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 25–36.
9. Yuki Ban, Takashi Kajinami, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2012b. Modifying an identified curved surface shape using pseudo-haptic effect. In *Haptics Symposium (HAPTICS), 2012 IEEE*. IEEE, 211–216.
10. Yuki Ban, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2014. Displaying shapes with various types of surfaces using visuo-haptic interaction. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*. ACM, 191–196.
11. Federico Barbagli, Ken Salisbury, Cristy Ho, Charles Spence, and Hong Z Tan. 2006. Haptic discrimination of force direction and the influence of visual information. *ACM Transactions on Applied Perception (TAP)* 3, 2 (2006), 125–135.
12. Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. 2009. Variational harmonic maps for space deformation. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 34.
13. Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. 2016. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 717–728.
14. Fred L. Bookstein. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence* 11, 6 (1989), 567–585.
15. Eric Burns, Sharif Razzaque, Abigail T Panter, Mary C Whitton, Matthew R McCallus, and Frederick P Brooks. 2005. The hand is slower than the eye: A quantitative exploration of visual dominance over proprioception. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*. IEEE, 3–10.
16. Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D Wilson. 2017. Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3718–3728.
17. Inrak Choi, Elliot W Hawkes, David L Christensen, Christopher J Ploch, and Sean Follmer. 2016. Wolverine: A wearable haptic interface for grasping in virtual reality. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 986–993.
18. Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation.. In *Uist*, Vol. 13. 417–426.
19. James J Gibson. 1933. Adaptation, after-effect and contrast in the perception of curved lines. *Journal of experimental psychology* 16, 1 (1933), 1.
20. Seth Copen Goldstein, Jason D Campbell, and Todd C Mowry. 2005. Programmable matter. *Computer* 38, 6 (2005), 99–101.
21. Xianfeng Gu, Feng Luo, Jian Sun, and S-T Yau. 2013. Variational principles for Minkowski type problems, discrete optimal transport, and discrete Monge-Ampere equations. *arXiv preprint arXiv:1302.5472* (2013).
22. Xianfeng David Gu and Shing-Tung Yau. 2008. *Computational conformal geometry*. International Press Somerville, Mass, USA.
23. Vincent Hayward, Oliver R Astley, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles-De-La-Torre. 2004. Haptic interfaces and devices. *Sensor Review* 24, 1 (2004), 16–29.
24. Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1957–1967.
25. Ken Hinckley, Randy Pausch, John C Goble, and Neal F Kassell. 1994. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 452–458.

26. Eberhard Hopf, Cathleen S Morawetz, James Serrin, and Sinaĭ. *Selected Works of Eberhard Hopf: With Commentaries*.
27. Brent Edward Insko, M Meehan, M Whitton, and F Brooks. 2001. *Passive haptics significantly enhances virtual environments*. Ph.D. Dissertation. University of North Carolina at Chapel Hill.
28. Luv Kohli. 2010. Redirected touching: Warping space to remap passive haptics. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*. IEEE, 129–130.
29. Luv Kohli, Mary C Whitton, and Frederick P Brooks. 2012. Redirected touching: The effect of warping space on task performance. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*. IEEE, 105–112.
30. Dimitrios A Kontarinis, Jae S Son, William Peine, and Robert D Howe. 1995. A tactile shape sensing and display system for teleoperated manipulation. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, Vol. 1. IEEE, 641–646.
31. Krzysztof Jakub Kruszyński and Robert van Liere. 2009. Tangible props for scientific visualization: concept, requirements, application. *Virtual reality* 13, 4 (2009), 235–244.
32. Eun Kwon, Gerard J Kim, and Sangyoon Lee. 2009. Effects of sizes and shapes of props in tangible augmented reality. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*. IEEE, 201–202.
33. Anatole Lecuyer, Sabine Coquillart, Abderrahmane Kheddar, Paul Richard, and Philippe Coiffet. 2000. Pseudo-haptic feedback: Can isometric input devices simulate force feedback?. In *Virtual Reality, 2000. Proceedings. IEEE*. IEEE, 83–90.
34. Susan J Lederman and Lynette A Jones. 2011. Tactile and haptic illusions. *IEEE Transactions on Haptics* 4, 4 (2011), 273–294.
35. Yongseok Lee, Inyoung Jang, and Dongjun Lee. 2015. Enlarging just noticeable differences of visual-proprioceptive conflict in VR using haptic feedback. In *World Haptics Conference (WHC), 2015 IEEE*. IEEE, 19–24.
36. Thomas H Massie, J Kenneth Salisbury, and others. 1994. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, Vol. 55. Chicago, IL, 295–300.
37. Yoky Matsuoka, Sonya J Allin, and Roberta L Klatzky. 2002. The tolerance for visual feedback distortions in a virtual environment. *Physiology & behavior* 77, 4 (2002), 651–655.
38. Theodor Meis and Ulrich Marcowitz. 2012. *Numerical solution of partial differential equations*. Vol. 32. Springer Science & Business Media.
39. Antonia-Roberto Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: ergonomic optimization of immersive virtual environments. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*. ACM.
40. Ken Nakagaki, Sean Follmer, and Hiroshi Ishii. 2015. Lineform: Actuated curve interfaces for display, interaction, and constraint. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 333–339.
41. Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. 2001. Redirected walking. In *Proceedings of EUROGRAPHICS*, Vol. 9. Manchester, UK, 105–106.
42. Gabriel Robles-De-La-Torre and Vincent Hayward. 2001. Force can overcome object geometry in the perception of shape through active touch. *Nature* 412, 6845 (2001), 445.
43. Irvin Rock and Jack Victor. 1964. Vision and touch: An experimentally created conflict between the two senses. *Science* 143, 3606 (1964), 594–596.
44. Anne Roudaut, Diana Krusteva, Mike McCoy, Abhijit Karnik, Karthik Ramani, and Sriram Subramanian. 2016. Cubimorph: designing modular interactive devices. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 3339–3345.
45. Anne Roudaut, Rebecca Reed, Tianbo Hao, and Sriram Subramanian. 2014. Changibles: analyzing and designing shape changing constructive assembly. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 2593–2596.
46. Jia Sheng, Ravin Balakrishnan, and Karan Singh. 2006. An interface for virtual 3D sculpting via physical proxy.. In *GRAPHITE*, Vol. 6. 213–220.
47. Walter A Strauss. 1992. *Partial differential equations*. Vol. 92. Wiley New York.
48. Ivan E Sutherland. 1965. The ultimate display. *Multimedia: From Wagner to virtual reality* (1965).
49. Christopher R Wagner, Susan J Lederman, and Robert D Howe. 2002. A tactile shape display using RC servomotors. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*. IEEE, 354–355.
50. Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin* 1, 6 (1945), 80–83.
51. Yiwei Zhao, Lawrence Kim, Ye Wang, Mathieu Le Goc, and Sean Follmer. 2017. Robotic Assembly of Haptic Proxy Objects for Tangible Interaction and Virtual Reality. In *International Conference on Interactive Surfaces and Spaces*. ACM.